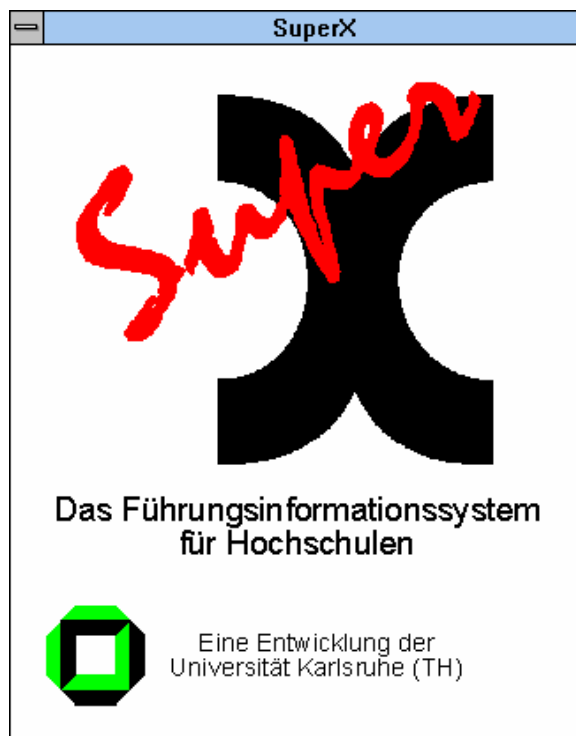


SuperX-Admin-Handbuch COB-Modul



www.MemText.de

- Daniel Quathamier
danielq@memtext.de
- Meikel Bisping
mbisping@memtext.de

Supportadresse

support@superx-projekt.de

Supportadresse speziell
für Baden-Württemberg:

support-bw@superx-projekt.de

<http://www.superx-projekt.de>

Version 1.0rc10

Stand 12.01.2010

Download als [PDF-Datei](#)

[Lehrfilm zur Installation des COB-Moduls](#)

Sun, Sun Microsystems, Solaris, Java, JavaServer Web Development Kit, JDBC und Java-Server Pages sind eingetragene Warenzeichen von Sun Microsystems, Inc. UNIX ist ein eingetragenes Warenzeichen von X/Open Company, Ltd. Windows, WindowsNT, Win32, VBScript und Office 2000 sind eingetragene Warenzeichen von Microsoft Corp. Linux ist eingetragenes Warenzeichen von Linus Torvalds. Alle weiteren Produktnamen sind Warenzeichen der jeweiligen Hersteller.

Dieses Produkt beinhaltet Software, die von der Apache Software Foundation (<http://www.apache.org/>) entwickelt wurde.

SuperX wird unter der deutschen Variante der GPL-Lizenz von dem Land Nordrhein-Westfalen, vertreten durch die FernUniversität Hagen, diese wiederum vertreten durch die Geschäftsstelle der Initiative CampusSource bei der FernUniversität Hagen, Feithstraße 142, D-58084 Hagen vertrieben (www.campussource.de). Details zu den Lizenzbedingungen finden Sie im COB-Modul-Archiv (/lizenz.txt) oder unter <http://www.campussource.de/lizenz/>. Ergänzende Hinweise finden Sie auf der Projekthomepage unter <http://www.superx-projekt.de>.

Inhaltsverzeichnis

1	Einführung.....	5
1.1	Hintergründe.....	5
1.2	Kurzüberblick.....	7
2	Installation des COB-Moduls.....	8
2.1	Kurzanleitung	9
2.2	Ordnerstruktur und Umgebung des COB-Moduls	11
2.3	Entladen der COB-Daten	12
2.3.1	DBMS-spezifische Konfiguration	12
2.3.1.1	Entladen aus Informix/ UNIX.....	12
2.3.1.2	Entladen aus Postgres/ Unix	13
2.3.1.3	Entladen aus Informix/ Windows	14
2.3.1.4	Entladen aus Access/ Windows	15
2.3.1.4.1	Entladen via RmiJdbc	15
2.3.1.4.2	Dateibasiertes Entladen.....	16
2.3.1.4.3	Rechteinstellungen in Access.....	18
2.3.2	Entladen mit oder ohne Gültigkeitszeiträumen	18
2.3.3	Nur beim PUSH-Verfahren: Kopieren der Rohdaten	19
2.4	Neuinstallation: Erzeugung der Prozeduren, Schlüssel und Datentabellen	19
2.5	Upgrade eines vorhandenen COB-Moduls	19
2.6	Laden der COB-Daten nach SuperX.....	20
2.6.1	Hochschulspezifische Transformationen.....	21
2.6.2	Separate Übernahme des Organigramms.....	21
2.6.3	Setzen von Texten für KLR-Geldgeber.....	22
2.6.4	Drittmittel	23
2.7	Vergeben von Rechten für eingeschränkte User	24
2.7.1	a. Institutionsrechte	24
2.7.2	b. Sichtrechte	24
2.7.2.1	Kostenstellen.....	24
2.7.2.2	Kostenträger.....	26
2.7.2.3	Kostenarten	26
2.8	Anpassungen der Berichte (nur relevant für NRW).....	26
2.8.1	Anpassungen des Berichtsblatts Kennzahlen aus der Kostenrechnung (nur NRW).....	27
2.9	Entfernen des COB-Moduls	27
3	Bestandteile des COB-Moduls.....	27
3.1	Datentabellen.....	28
3.1.1	Tabelle cob_busa: Primärbuchungen.....	28
3.1.2	Tabelle cob_vtbu: Verteilbuchungen.....	29
3.1.3	Tabelle cob_umks: Verzeichnis der Verrechnungssätze/Festpreise	29
3.1.4	Tabelle cob_su_imp_stud: Studierendendaten in COB	29
3.2	Schlüsseltabellen	29
3.2.1	Die Tabelle cifx	29
3.2.2	Die Tabellen cob_cif / cob_cifx	30
3.2.3	Die Tabelle cob_inst (Verzeichnis der Kostenstellen)	30
3.2.4	Die Schlüsseltabelle cob_fikr (Verzeichnis der Kosten-/Erlösarten)	31
3.2.5	Die Tabelle cob_proj (Verzeichnis der Kostenträger/Projekte).....	31
3.2.6	Die Tabelle cob_exkotr (Verzeichnis der externen Kostenträger).....	31

3.2.7	Die Schlüsseltabelle cob_kpum (Verzeichnis der Leistungsarten).....	31
3.2.8	Die Schlüsseltabelle cob_proj_to_inst.....	32
3.2.9	Die Schlüsseltabelle aggregierung.....	32
3.2.10	Die Schlüsseltabelle cob_drittmittel (obsolet).....	32
3.2.11	Die Schlüsseltabelle cob_stug (Verzeichnis der Studiengänge).....	32
3.2.12	Die Schlüsseltabelle cob_geldgeber	32
3.2.13	cob_alt_hier, cob_trees, cob_tree_cfg, cob_alt_keys	33
3.3	Hilfstabellen	33
3.3.1	cob_busa_aggr.....	33
3.3.2	cob_vtbu_aggr	34
3.3.3	cob_busa_vtbu.....	34
3.4	Alternative Hierarchien / Sichten	34
3.4.1	Prozedur sp_cob_alt_hier	35
3.4.2	Prozedur sp_cob_inst_hier	35
3.4.3	Prozedur sp_cob_ktr_hier.....	35
3.4.4	Prozedur sp_cob_fikr_hier	35

1 Einführung

Das Berichtssystem SuperX ist ein sog. Data Warehouse, d.h. beliebig viele Datenquellen werden unter einer einheitlichen Auswertungsschnittstelle zur Verfügung gestellt. Da jede Hochschule unterschiedliche Datenquellen besitzt und nach SuperX übernehmen will, bereiten wir für jede Datenquelle ein Modul vor, z.B. ein COB-Modul oder ein SOS-Modul.

Das SuperX-COB-Modul entlädt Daten aus HISCOB (Version 5.x-11.x) und lädt sie in SuperX, um dort Abfragen im Bereich Kostenrechnung zu ermöglichen.

Das COB-Modul wurde entsprechend der Anforderungen aus dem SuperX-Projekt in Baden-Württemberg erweitert. Erste Tests des Moduls an Pilothochschulen haben gezeigt, dass der erste Aufbau von Abfragemasken bei Benutzung umfangreicher alternativer Hierarchien recht lange dauert. Außerdem wurde der Wunsch geäußert, dass ein eingestelltes Standarddatum von einer Abfrage zur nächsten beibehalten wird.

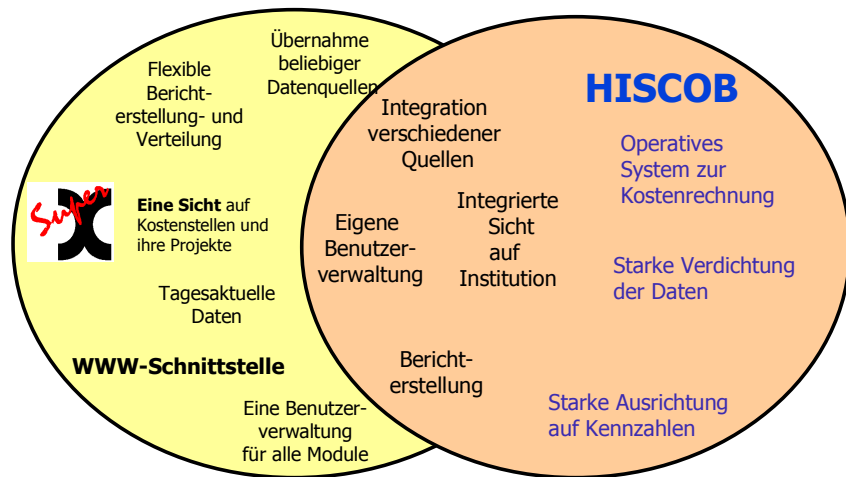
Wir werden versuchen, in den Folge Releases Verbesserungen in diesen Punkten zu erreichen.

1.1 Hintergründe

Das COB-Modul dient als WWW-basiertes Frontend für HISCOB, das Controlling-System der HIS GmbH. Es liefert Daten aus HISCOB über eine einheitliche Benutzerschnittstelle aus und kann z.B. für dezentrale User via WWW zur Verfügung gestellt werden. Die Berichte in SuperX bilden teilweise vorhandene Berichte in HISCOB nach (z.B. das Schnellinfo), teilweise werden andere Akzentuierungen vorgenommen. Die vorhandenen Abfragen dienen als Beispiele und können von den Anwendern beliebig an örtliche Gegebenheiten angepasst werden.

Sowohl SuperX als auch COB sind Systeme, in denen Daten aus anderen Systemen zusammengestellt und verdichtet werden. Zunächst stellt sich daher die Frage, welche zusätzlichen Funktionen SuperX bietet. Die folgende Abbildung stellt beide Systeme zusammen:

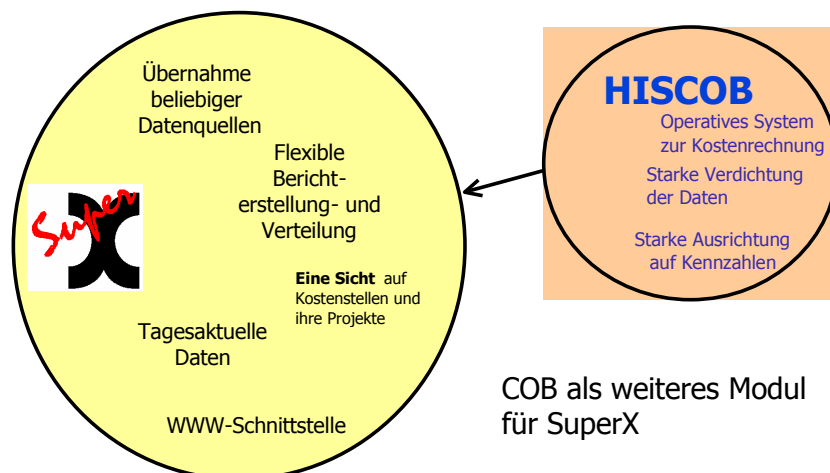
HISCOB besitzt als Controlling-Instrument eine starke Ausrichtung auf Kennzahlen, während SuperX eher deskriptiv ausgerichtet ist. Die Datenquellen für COB sind festgelegt, in SuperX können beliebige Datenquellen aufgenommen werden. Beide Systeme bieten die Möglichkeit zur Berichterstellung, SuperX auch via WWW.



Anders als in HISCOB können die Statistiken in SuperX je nach Modul stichtagsbezogen oder tagesaktuell sein, während sie in HISCOB eher stichtagsbezogen sind.

Die folgende Abbildung veranschaulicht die Integrationsmöglichkeit der beiden Systeme. Dabei wird HISCOB als zusätzliche Datenquelle in ein umfassenderes System SuperX aufgenommen.

Wenn ein umfassendes Data Warehouse mit mehreren Modulen (z.B. SOS, SVA) existiert, kann COB als weiteres Modul aufgenommen werden. Ggf. müssten dann allerdings gemeinsame Schlüssel angepasst werden.

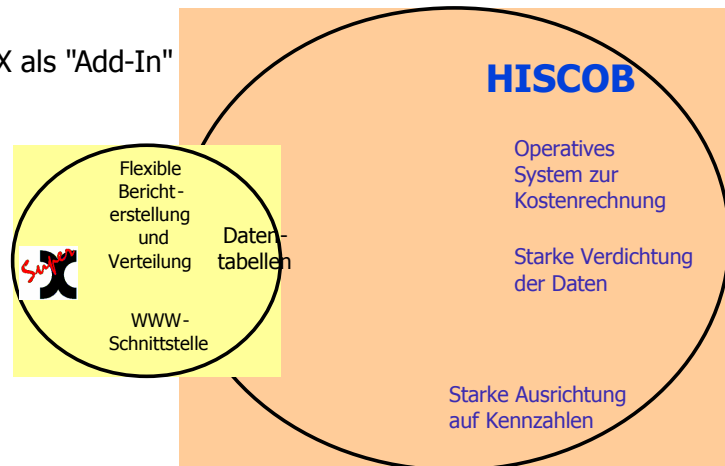


Außerdem muss ein Entladerhythmus aus COB und ein Übernahmerrhythmus nach SuperX eingestellt werden.

Grundsätzlich besteht noch eine weitere, „abgeschwächte“ Integrationsmöglichkeit, bei der SuperX (d.h. Kernmodul + COB-Modul) ein zusätzliches Frontend für die COB-Datenbank wird.

SuperX als Add-In: Es liefert eigene Berichte direkt aus COB, und enthält keine anderen Module. Das SuperX-Kernmodul wird zusammen mit dem COB-Modul auf dem COB-Server (in einer getrennten Datenbank) installiert, und die Datentabellen werden direkt (d.h. ohne Entladen) aus COB übernommen.

SuperX als "Add-In"



Diese Variante wird momentan nicht durch automatisierte Skripte unterstützt.

1.2 Kurzüberblick

Die Module enthalten die wichtigsten Prozeduren, Tabellen und Abfragen für die jeweilige Datenquelle. Folgende Tabellen sind generell zu unterscheiden:

- *Datentabellen* enthalten die entladenen Basisdaten aus HISCOB
- *Hilfstabellen* enthalten aggregierte Tabellen und werden von den Abfragen genutzt. Durch Hilfstabellen wird die Performance der Abfragen besser, außerdem stehen bei möglichen Ladefehlern die relevanten Tabellen für die Abfragen noch zur Verfügung.
- *Schlüsseltabellen* enthalten Schlüssel und Metadaten, z.B. Kostenarten, Semester, Abschlüsse etc.

Das COB-Modul besteht im Endzustand aus Tabellen, Prozeduren und Abfragen. Da die COB-Datenbank der HIS-GmbH an sehr vielen Hochschulen eingesetzt wird, ist das COB-System gut geeignet für die Übernahme nach SuperX. Das COB-Modul ist weniger komplex als z.B. das SOS-Modul.

Falls es bei der Implementation des COB-Moduls zu Problemen kommt, können Sie sich unter www.superx-projekt.de informieren.

Supportadresse allgemein:
support@superx-projekt.de
Supportadresse für Baden-Württemberg:
support-bw@superx-projekt.de

Sven Gutow gutow@his.de	Daniel Quathamier danielq@memtext.de
Oliver Seegers seegers@his.de	Meikel Bisping mbisping@memtext.de

2 Installation des COB-Moduls

Um das COB-Modul zu installieren, muss man zunächst das Kernmodul Version 3.0(rcx) von SuperX installieren. Dies lässt sich auf einem separaten Rechner installieren, aber auch auf dem COB-Rechner, je nach Installationsvariante.

Zunächst gehen Sie in das Verzeichnis `$SUPERX_DIR` (normalerweise `/home/superx`) und entpacken dort das Archiv `cob<<Versionsnr>>.tar.gz`.

Bei der Installation des COB-Moduls werden zentrale Schlüsseltabellen erzeugt und Installationsscripte gestartet. Zunächst müssen die notwendigen Tabellen erzeugt werden, danach können Daten aus COB übernommen werden. Im Anschluss daran werden Hilfstabellen erzeugt, und die Abfragen eingespielt. Die Leserechte müssen dann manuell gepflegt werden. Damit zukünftig COB-Daten in die SuperX-Datenbank eingelesen und mitgelieferte Abfragen getestet werden können, muss die SuperX-Datenbank zunächst einmalig um die COB-Bestandteile erweitert werden. Dazu gehören Tabellen (Basisdaten, Schlüsseltabellen), Prozeduren, Abfragen und Masken.

Die Scripte des COB-Moduls laufen unter LINUX (Suse, RedHat - ggfs. nach gewissen Anpassungen auch unter AIX und SOLARIS) und unter Windows mit Cygwin.

2.1 Kurzanleitung

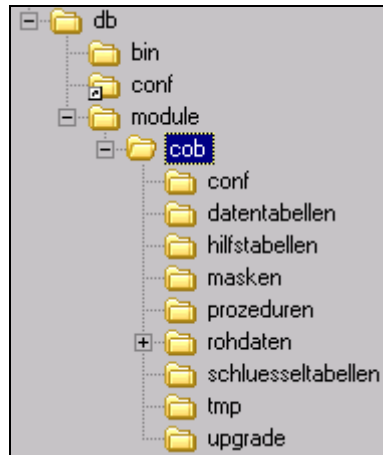
Folgende Arbeitsschritte sind notwendig:

1. Entpacken des COB-Moduls in `$SUPERX_DIR` und Einrichten der Umgebung (Prüfen ob in Ihrer `$SUPERX_DIR/db/bin/SQL_ENV` alle Einträge aus `SQL_ENV.cob.sam` vorhanden sind, ggfs. rüberkopieren). Aktivieren Sie die Umgebung mit
`. $SUPERX_DIR/db/bin/SQL_ENV`
2. Wenn der COB-Rechner ein Windows-Rechner ist, müssen Sie zunächst den SuperX-Client installieren (<http://download.superx-projekt.de>); dies ist im Paket des SuperX-Client dokumentiert.
3. Kopieren der Entladescripte unter `$SUPERX_DIR/db/module/cob/rohdaten` zum COB-Rechner
4. Anpassen der **Umgebungsvariablen** (S. 12) für das Entladescript (COB-Version, Startjahr) in der Datei `$SUPERX_DIR/db/module/cob/rohdaten/COB_ENV` (für UNIX) bzw. `cob_env.bat` (Windows). Für beide Dateien sind Muster mit jeweils der Endung `.sam` vorhanden.
5. Nur unter Windows: Entscheiden Sie sich für eine Entladevariante (RMI-ODBC oder dateibasiertes Entladen) und gehen Sie entsprechend der Anleitung vor.
6. Ausführen des Entladescripts `cob_unload.x` (UNIX) bzw. `cob_unload.bat` (Windows). Prüfen Sie die Logdatei `cob_unload.err`
7. Kopieren des Verzeichnisinhalts von `rohdaten` vom COB-Rechner nach
`$SUPERX_DIR/db/module/cob/rohdaten`
8. Erzeugen des **COB-Moduls** (S. 19) in der SuperX-Datenbank:
`$SUPERX_DIR/db/module/cob/cob_modul_erzeugen.x`
Es werden Tabellen, Views etc. erzeugt, und die Masken werden eingespielt. Nach der Installation haben die User der Gruppen *Administration* oder *superx* Leserecht auf die Abfragen. Falls ein Fehler auftritt, versuchen Sie die Ursache zu beheben, starten Sie dann
`$SUPERX_DIR/db/module/cob/cob_modul_entfernen.x` (etwaige Fehler ignorieren)
und anschließend erneut
`$SUPERX_DIR/db/module/cob/cob_modul_erzeugen.x`
9. Nun bereiten Sie die erste Übernahme vor. Je nach Datenumfang und Rechner bietet es sich an, am Anfang nur wenige Rohdaten zum Testen zu nehmen, sonst dauert der erste Update je nach Datenmenge 1-6 h. Dazu gibt es ein Shellsript `cob_shrink.x`, das die Rohdaten auf max. 100 Sätze kürzt. Beim Echtbetrieb später können Sie mit `cob_unshrink.x` dies wieder rückgängig machen.
.
10. Um Probleme der Vergleichbarkeit mit HISCOB aufgrund von Gültigkeitszeiträumen zu vermeiden, sollte die Gültigkeit von Kostenstellen,-arten und -trägern überschrieben und die Standbuttons für die Sichten abgeschaltet werden. Erstellen Sie dazu eine Kopie der Datei `finalize.sql.sam` und nennen Sie sie `finalize.sql`
11. Übernahme der entladenen **COB-Daten** (S. 20) nach SuperX:
`$SUPERX_DIR/db/module/cob/cob_update.x --mit-organigramm`
(falls Sie das Organigramm anwederweitig pflegen, lassen Sie den Parameter `--mit-organigramm` weg)
Die Logdatei heißt `cob_update.err`.

12. Bei Fehlern oder Warnungen im Update wird eine Meldung in der Logdatei ausgegeben. Ansonsten melden Sie sich im Applet neu an, und testen Sie nun die Abfragen.
(Im XML-Frontend müssen Sie zunächst unter <http://rechnernamen:8080/superx/servlet/SuperXManager> den Server-Cache aktualisieren und sich dann neu anmelden, oder Sie starten Tomcat einfach neu)
13. Für eingeschränkte User müssen Sie Rechte vergeben (vergl. entsprechender Abschnitt)
14. Schritt 10 wird bei jedem SuperX-Update wiederholt. Nun muss der Entladerhythmus geplant werden, und die Cronjobs werden eingerichtet. Eine Vorlage befindet sich in `$COB_PFAD/cob_update_cron.x.sam`

2.2 Ordnerstruktur und Umgebung des COB-Moduls

Die folgende Abbildung zeigt die Ordnerstruktur des COB-Moduls.



Das Masken-Verzeichnis im COB-Modul ist nicht zu verwechseln mit dem des Kernmoduls: Im Masken-Verzeichnis des COB-Moduls werden die bei Installation mitgelieferten COB-Abfragen gespeichert; das Masken-Verzeichnis des Kernmoduls dient als Arbeitsbereich für eigene Anpassungen. Diese Trennung ist wichtig, falls Sie Updates oder neue Abfragen zum COB-Modul installieren wollen.

Die jeweiligen Pfade zum COB-Modul werden in der Datei `$SUPERX_DIR/db/bin/SQL_ENV` festgelegt. Übertragen Sie ggf. die Angaben von `db/bin/SQL_ENV_cob.sam` nach

`$SUPERX_DIR/db/bin/SQL_ENV`, und rufen Sie das Script einmal auf mit

`. $SUPERX_DIR/db/bin/SQL_ENV`

Die folgende Tabelle zeigt einen Auszug aus der `SQL_ENV`

```
SUPERX_MODULE=$SUPERX_DIR/db/module; export SUPERX_MODULE
SUPERX_ROHDATEN=rohdaten; export SUPERX_ROHDATEN

COB_PFAD=$SUPERX_MODULE/cob; export COB_PFAD
COB_ERRORDAT=$COB_PFAD/cob_update.err; export COB_ERRORDAT
COB_LOAD_PFAD=$COB_PFAD/$SUPERX_ROHDATEN; export COB_LOAD_PFAD
COB_ERRORMAIL=$ERRORMAIL #Standard-Mailadr.
export COB_ERRORMAIL
COB_LOGMAIL=$LOGMAIL #Standard-Mailadr.
export COB_LOGMAIL
COB_BACKUP="false"
export COB_BACKUP
```

Sie müssen diese Pfade ergänzen, damit die Scripte und cron-Jobs laufen.

In den jeweiligen Scripten wird diese für das Setzen der Umgebungsvariablen genutzt, so dass in der `.profile` des Users SuperX keine Änderungen notwendig sind. Hinweis für Datenbankserver unter AIX oder anderen Linux / Unix-Derivaten: Beachten Sie, dass die Scripte nur dann lauffähig sind, wenn auf dem Datenbankserver unter `/bin` die Datei (oder ein Link) `sh` und `bash` liegt. Die Scripte von SuperX erwarten die bash-Shell Linux-konform im Ver-

zeichnis /bin; wenn dies nicht der Fall ist, sollten `sh` und `bash` z.B. von `/usr/bin` nach `/bin` kopiert oder gelinkt werden.

2.3 Entladen der COB-Daten

Das Entladen aus COB gestaltet sich je nach Datenbank- und Betriebssystem unterschiedlich. Zunächst wird der häufigste und einfachste Fall beschrieben, das Entladen unter UNIX.

Kopieren Sie die Datei

```
$SUPERX_DIR/db/module/cob/rohdaten/COB_ENV.sam
```

nach

```
$SUPERX_DIR/db/module/cob/rohdaten/COB_ENV
```

und passen Sie sie entsprechend der weiteren Anleitung an.

COB_ENV Die Umgebung für das Entladescript aus COB

```
##Pfad für Entladedaten:
COB_PFAD=. ; export COB_PFAD
##hier muss Unterverzeichnis unl existieren!
DBASE=cob
export DBASE
#Start des Ladens:
start_cob=2002; export start_cob #Startzeitpunkt
VERSION=11
export VERSION
```

Im Kopf der Datei `cob_unload.x` wird diese Datei aufgerufen; zunächst muss das Ausgabeverzeichnis `COB_PFAD` und das Startjahr, ab dem COB-Daten vorliegen bzw. entladen werden sollen, angegeben werden. Außerdem müssen der COB-Datenbankname (`DBASE`) und die Versionsnummer eingegeben werden (Ganzzahl).

Zum Entladen via "Push" oder "Pull" siehe das SuperX-Adminhandbuch Kernmodul.

Zunächst werden DBMS-spezifische Parameter beschrieben. Darüberhinaus können Sie auch steuern, ob die [Gültigkeiten](#) zentraler Schlüsseltabellen ignoriert werden sollen oder nicht.

2.3.1 DBMS-spezifische Konfiguration

2.3.1.1 Entladen aus Informix/ UNIX

Einige Tabellen der COB -Datenbank werden durch ein Entladescript entladen. Das Script befindet sich hier:

```
$SUPERX_DIR/db/module/cob/rohdaten/cob_unload.x
```

Wenn Sie im Push-Verfahren entladen: Kopieren Sie den gesamten Verzeichnisinhalt ab `/rohdaten` auf den COB-Rechner, geben Sie dem Script `cob_unload.x` Ausführungsrechte, ko-

pieren Sie die Beispieldatei `COB_ENV.sam` nach `COB_ENV` und passen Sie darin die Umgebungsvariablen an.

COB_ENV Informix

```
DATABASE=INFORMIX; export DATABASE #Datenbanksystem  
VERSION=11; export VERSION #HIS-Modulversion
```

Im Kopf der Datei `cob_unload.x` wird diese Datei aufgerufen; hier muss `INFORMIXSERVER` und `INFORMIXDIR` angegeben werden.

Testen Sie zunächst den Zugriff mit dem Befehl:

```
. COB_ENV  
dbaccess $DBASE
```

Wenn die Datenbankverbindung klappt, dann starten Sie das Script mit `cob_unload.x`

Die Rohdaten werden in das Unterverzeichnis `rohdaten/unl` kopiert. Das Entladedatum wird danach in der Textdatei `rohdaten/superx.datum` gespeichert; wenn das Script einen Fehler findet, dann wird das vorherige Datum (in der Datei `superx.datum.alt`) gesetzt.

Ausgaben des Scriptes werden in die Datei `cob_unload.err` geschrieben. Diese Datei wird auf Meldungen über Entladefehler geprüft.

Wenn Sie das Organigramm aus COB benutzen und öfter aktualisieren wollen als das gesamte COB-Modul, haben wir ein Entladescript nur für das Organigramm (und verwandte Tabellen) erzeugt. Dies liegt ebenfalls im Verzeichnis `$COB_LOAD_Pfad` und lautet `inst_unload.x`

Die Umgebungssteuerung und der Start des Scripts funktioniert analog zum normalen COB-Unload.

2.3.1.2 Entladen aus Postgres/ Unix

Wenn Sie HISCOB mit Postgres einsetzen müssen Sie wie im vorherigen Abschnitt beschrieben die `COB_ENV.sam` nach `COB_ENV` kopieren und anpassen. `SX_CLIENT` muss `psql` sein.

Wir empfehlen, den Entladevorgang vom SuperX-Rechner aus zu starten ("Pull"-Verfahren):

Zum Zugriff auf die HISCOB-Datenbank müssen Sie analog zur `db.properties` die `db-cob.properties` anpassen, nachdem Sie die Beispieldatei `db-cob.properties.sam` nach `db-cob.properties` kopiert haben. Verwenden Sie `propadmin.x /path/to/cob/db-cob.properties`. Wenn Sie mit COB-GX 12 oder höher arbeiten, wird als Datenbankname nicht mehr "cob" angegeben, sondern "hism". Eine Musterdatei `db-hism_pg.properties.sam` liegt im Archiv.

Zum Entladen verwenden Sie das Script `cob_unload.x`.

Wenn Sie den Entladevorgang auf dem COB-Rechner durchführen wollen ("Push-Verfahren"), müssen Sie dort Java 1.4.2 oder höher installieren. Außerdem werden Bibliotheken aus dem SuperX-Kernmodul benötigt.

Sie können das Kernmodul z.B. unter `home/cob/superx` entpacken. Erstellen Sie eine `SQL_ENV` in der dann als `$SUPERX_DIR=/home/cob/superx` eingetragen ist.

Vor dem Start des Entladevorgangs müssen immer die Umgebungsvariablen in der `SQL_ENV` geladen werden mit `. /home/superx/cob/db/bin/SQL_ENV`.

Im Weiteren verfahren Sie analog zur Vorgehensweise beim Entladen aus Informix/Unix, nur dass natürlich `INFORMIXSERVER` etc nicht angegeben werden braucht. `PGHOST` oder `PGPORT` wird ebenfalls im COB-Modul nicht benötigt, da der Unload intern mit `jdbc` läuft. Weitere Variablen in der Datei `COB_ENV`:

JDBC_PARAM	Wenn Sie unter Postgres aus COB-GX 12 entladen, muss beim Unload ein spezielles JDBC-Kommando abgesetzt werden, das dem Client sozusagen den Weg zum SVA-Schema zeigt. Dieses Kommando wird wie folgt aktiviert: <pre>JDBC_PARAM="set search_path to cob;" export JDBC_PARAM</pre>
JDBC_CLASSPATH	Wenn Sie den obigen <code>JDBC_PARAM</code> nutzen, aber noch nicht mit dem Kernmodul 4.0 arbeiten, müssen Sie eine spezielle Bibliothek laden mit dem Befehl <pre>JDBC_CLASSPATH="\$COB_LOAD_PFAD/lib/superx3.9.jar:\$JDBC_CLASSPATH" export JDBC_CLASSPATH</pre> Wenn Sie bereits mit dem Kernmodul 4.0 arbeiten, ist dies nicht nötig.

2.3.1.3 Entladen aus Informix/ Windows

Wenn Sie Informix unter Windows betreiben, können Sie das Entladen vom SuperX-Rechner aus durchführen.

Setzen Sie in der Datei

```
$SUPERX_DIR/db/module/cob/rohdaten/COB_ENV
```

die Variable `SX_CLIENT=jdbc` und passen Sie alle benötigten Umgebungsvariablen an.

Kopieren Sie `db-cob.properties.sam` nach `db-cob.properties` und tragen Sie die Verbindungsparameter zur Informixdatenbank mit dem Propadmin ein.

```
propadmin.x /path/to/cob/db-cob.properties
```

Der Entladevorgang wird gestartet mit

```
cob_unload.x
```

Die Rohdaten werden in das Unterverzeichnis `rohdaten/unl` kopiert. Das Entladedatum wird danach in der Textdatei `rohdaten/superx.datum` gespeichert; wenn das Script einen Fehler findet, dann wird das vorherige Datum (in der Datei `superx.datum.alt`) gesetzt.

Ausgaben des Scriptes werden in die Datei `unload.err` geschrieben. Diese Datei wird auf Meldungen über Entladefehler geprüft.

2.3.1.4 Entladen aus Access/ Windows

Wenn Sie aus Access entladen wollen, ist das Vorgehen etwas anders, weil Access keine Serverapplikation ist.

Zunächst müssen Sie Java ab 1.4.2 (besser 1.5) und das SuperX-Client 2.1-Paket auf dem Windows-Rechner installieren. Eine Anleitung findet sich im Administratorhandbuch des Kernmoduls.

Je nach Server-Konfiguration entladen Sie die Daten vom SuperX-Rechner aus unter UNIX, indem Sie via [RmiJdbc](#) (S. 15) vom SuperX-Rechner auf einen entfernten COB-Server zugreifen. Wir empfehlen dieses Vorgehen, weil es sich besser automatisieren lässt. Alternativ können Sie auf dem Windows-Rechner aus COB entladen und die Rohdaten auf den SuperX-Datenbankserver kopieren ("Dateibasiertes Entladen").

2.3.1.4.1 Entladen via RmiJdbc

Der Rmi-JDBC-Treiber der Firma ObjectWeb, der Teil des SuperX-Clientpackets ist, macht aus Access einen Datenbankserver. Zunächst müssen Sie auf dem Windows-Rechner eine Applikation starten, die die ODBC-Verbindung als Serverapplikation verfügbar macht.

Nachdem Sie das SuperX-Clientpaket (<http://download.superx-projekt.de>) installiert haben, starten Sie den RMI-Server in der DOS-Box mit `$SUPERX_DIR\db\bin\start_rmiodbc_server.bat`.

Danach können Sie vom SuperX-Rechner mit dem Propadmin darauf zugreifen.

Kopieren Sie `$SUPERX_DIR\db/module/cob/rohdaten/COB_ENV.sam` nach

`COB_ENV` und bearbeiten Sie die Verbindungsparameter.

`SX_CLIENT` muss `jdbc` sein, `DATABASE = ACCESS`.

Rufen Sie die `COB_ENV` einmal auf mit:

```
. COB_ENV
```

Kopieren Sie die `db-cob-odbc.properties.sam` nach `db-cob.properties`.

Starten Sie

```
propadmin.x $COB_LOAD_PFAD/db-cob.properties
```

Falls noch nicht automatisch ausgewählt, wählen Sie den Datenbanktreiber `rmiOdbc`; die Verbindungsparameter sind dabei nach dem folgenden Muster anzugeben:

```
jdbc:rmi://<rmiHostName[:port]>/<jdbc-url>
```

Wenn Sie zum Beispiel auf den Windows-Rechner mit dem Namen `cobserver` zugreifen wollen und dort COB unter Access installiert ist, und die `odbc`-Quelle dort `cob7` heißt, dann lautet der Pfad:

```
jdbc:rmi://cobserver/jdbc:odbc:cob7
```

Mit "Verbindung testen" können Sie prüfen, ob die Datenbank erreichbar ist, und dann speichern Sie die Datei. Wenn die Meldung kommt, dass die Tabelle `xdummy` nicht existiert, war der Verbindungsaufbau erfolgreich – lediglich die Tabelle wurde nicht gefunden, was aber

unproblematisch ist. Bei Recheproblemen beachten Sie den Abschnitt zu Rechtevergabe in Access.

Wenn der RMI-Server permanent laufen soll, müssen Sie die Sicherheitshinweise der Firma [ObjectWeb](#) beachten.

Danach können Sie den Entladeprozess vom SuperX-Rechner aus starten mit

```
cob_unload.x
```

Die Rohdaten werden in das Unterverzeichnis `rohdaten/unl` kopiert. Das Entladedatum wird danach in der Textdatei `rohdaten/superx.datum` gespeichert; wenn das Script einen Fehler findet, dann wird das vorherige Datum (in der Datei `superx.datum.alt`) gesetzt.

Ausgaben des Scriptes werden in die Datei `unload.err` geschrieben. Diese Datei wird auf Meldungen über Entladefehler geprüft.

2.3.1.4.2 Dateibasiertes Entladen

Wenn Sie das SuperX-Clientpaket installiert haben (<http://download.superx-projekt.de>), dann kopieren Sie das Verzeichnis `rohdaten` im COB-Modul in ein Verzeichnis des Windows-Clients, wenn möglich unter Einhaltung der Modulstruktur in SuperX; wenn Sie den Client z.B. nach `c:\superxclient\` installiert haben, dann kopieren Sie das Verzeichnis `rohdaten` aus dem COB-Modul nach `c:\superxclient\db\module\cob`

Kopieren Sie die Datei `c:\superxclient\db\bin\sql_env.bat.sam` nach `sql_env.bat` und passen Sie die Umgebungsvariablen an.

Öffnen Sie eine Shell (Eingabeaufforderung) und rufen Sie im Verzeichnis `c:\superxclient\db\bin`

die Datei

```
sql_env.bat
```

auf.

Als Nächstes wechseln Sie ins Verzeichnis `c:\superxclient\db\module\cob\rohdaten` und kopieren `cob_env.bat.sam` nach `cob_env.bat` und passen darin die Umgebungsvariablen an.

Rufen Sie im Verzeichnis `rohdaten` die `cob_env.bat` auf.

Kopieren Sie nun die Datei `db-cob.properties.sam` nach `db-cob.properties`.

Nun bearbeiten Sie die Verbindungsparameter mit

```
propadmin.bat
```

DB-Properties Admin 3.0

DB-Properties Admin für .\db-cob.properties

Driver : **JDBC-ODBC Bridge from Sun** (mögliche Datenbanksysteme für SuperX)

Driver Class: (muss im CLASSPATH stehen!)

Connection URL:

Username: Password:

Log Level SQL: **WARNING**

Log Level XML: **WARNING**

Masken, die im Cache sein sollen:

User, die im Cache sein sollen:

Der Apache ConnectionPool verwaltet die Anzahl benötigter Verbindungen dynamisch.
min/max idle gibt an wieviele Connections ständig bereit gehalten werden sollen.
maxActive gibt an wieviele Connections maximal gleichzeitig aktiv sein sollen.

minIdle: maxIdle: maxActive:

Verbindung testen **Speichern**

Wenn Sie COB unter Access betreiben, wählen Sie als Treiber die JDBC-ODBC-Bridge von Sun, und tragen bei dem Feld "connection url" den Namen der odbc-Quelle ein, normalerweise `cob`. Bitte beachten Sie hierbei, dass Sie unter Access auch ein Kennwort zuweisen müssen, d.h. der Standardbetrieb von COB im Auslieferungszustand mit Username `cob` und ohne Passwort ist für SuperX nicht möglich. Der Grund liegt darin, dass SuperX Passwörter verschlüsselt speichert (vergl. auch Abschnitt zu Rechtevergabe in Access).

Der Access-jdbc-Treiber befindet sich automatisch im CLASSPATH, er ist Teil der Java-Runtime von SUN.

Mit "Verbindung testen" können Sie prüfen, ob die Datenbank erreichbar ist, und dann speichern Sie die Datei. Wenn die Meldung kommt, dass die Tabelle `xdummy` nicht existiert, war der Verbindungsaufbau erfolgreich – lediglich die Tabelle wurde nicht gefunden, was aber unproblematisch ist.

Dann starten Sie das Entladen aus der DOS-Box mit "`cob_unload.bat`". Wenn alles klappt, dann liegt die Protokolldatei in `cob_unload.err`, und die Dateien im Unterverzeichnis "`unl`".

Prüfen Sie, ob `cob_unload.err` Fehlermeldungen enthält; wenn nein, dann kopieren Sie das gesamte Verzeichnis `$SUPERX_DIR/db/module/cob/rohdaten` auf den SuperX-Rechner.

Auch hier gibt es wie beim Entladen unter UNIX die Möglichkeit, nur die `inst`-Tabelle (und verwandte Tabellen) zu entladen, das Script dazu heißt `inst_unload.bat`.

Der ganze Entladevorgang lässt sich unter Windows (je nach Windows-Version) mit dem `net`-Befehl automatisieren, wichtig ist aber, dass vorher die Dateien `sql_env.bat` unter `db\bin` und `cob_env.bat` unter `db\module\cob\rohdaten` durchlaufen werden. Für das Kopieren auf den SuperX-Rechner bietet sich eine Fileserver-Lösung an: Sie mounten das Verzeichnis `$SUPERX_DIR/db/module/cob/rohdaten` als Samba-Share mit einem Laufwerksbuchstaben auf

dem Windows-Rechner, und starten das Programm in diesem Verzeichnis. So braucht man sich nicht um das Kopieren der Dateien zu kümmern.

2.3.1.4.3 Rechteinstellungen in Access

Wenn Sie (verständlicherweise) die schräge Benutzerverwaltung von MS Access nicht ausreizen wollen, empfehlen wir, einfach die Datenbank in Access zu öffnen und ein Datenbankkennwort zuzuweisen. Dass dieses Kennwort laut Access für den User `Administrator` und nicht für `COB` gilt, gehört zu den Mysterien des Access-Universums. Wir hatten auch schon Fälle, wo wir zwar einen Connect zu Datenbank machen konnten, aber keine Tabelleninhalte lesen durften. Kurzum: Eine funktionierende jdbc-Zugang erhalten Sie, wenn Sie der Datenbank ein Kennwort zuweisen, und mit dem User `cob` und dem besagten Kennwort eine Anmeldung machen

2.3.2 Entladen mit oder ohne Gültigkeitszeiträumen

Die Entladeroutinen erlauben es, **Gültigkeitszeiträume** von integrierten Schlüsseltabellen (`inst`, `fikr`, `gege`) zu ignorieren. Dies kann in folgendem Fall nützlich sein:

COB-GX ignoriert die **Gültigkeit von Schlüsseln**, SuperX aber nicht. Wenn Sie also Gültigkeitszeiträume begrenzt haben, kann es vorkommen dass Berichtsergebnisse in SuperX anders aussehen als in COB-GX. Um dies zu vermeiden können Sie die Gültigkeiten beim Entladen aus "Unendlich" setzen. Dies ist bei der Auslieferung des SuperX-COB-Moduls sogar der Standard.

Um die Gültigkeitszeiträume zu ignorieren gehen Sie bitte wie folgt vor:

COB_ENV Gültigkeiten ignorieren jeweils mit '0'	<pre>#Gültigkeiten ignorieren? Wenn ja, dann '0' wählen, wenn nein, dann '' wählen #Default ist '0', d.h. die Gültigkeit wird ignoriert. #Tabelle inst: COB_VONBIS_INST='' export COB_VONBIS_INST #Tabelle fikr: COB_VONBIS_FIKR='' export COB_VONBIS_FIKR #Tabelle gege: COB_VONBIS_GEGE='' export COB_VONBIS_GEGE</pre>
--	---

Die Schalter lassen sich jeweils für die Tabellen `inst`, `fikr` und `gege` steuern. Sie werden vom Entladescript `cob_unload.x` und `inst_unload.x` ausgewertet.

2.3.3 Nur beim PUSH-Verfahren: Kopieren der Rohdaten

In der Datei `COB_ENV.sam` sind noch weitere Umgebungsvariablen definiert (`REMOTE_HOST` etc). Diese werden von dem Script `cob_copy.x` benutzt, das die entladenen Rohdaten automatisch auf den SuperX-Rechner kopiert. Als unterstützte Übertragungsmethoden gelten dabei `scp` und `rsync`. Eine Anleitung zum Kopieren mit `scp/rsync` befindet sich im Adminhandbuch Kernmodul.

2.4 Neuinstallation: Erzeugung der Prozeduren, Schlüssel und Datentabellen

Diese Schritte brauchen nur einmal ausgeführt zu werden.

Unter UNIX:

1. Melden Sie sich als Benutzer `superx` an und wechseln Sie ins Verzeichnis

```
$SUPERX_DIR/db/module/cob.
```

2. Geben Sie ggf. dem Skript `cob_modul_erzeugen.x` (sowie allen anderen Scripten in diesem Verzeichnis) Ausführungsrechte mit

```
chmod +x *.x.
```

3. Starten Sie das Skript durch Eingabe von `cob_modul_erzeugen.x`.

Wenn ein Fehler aufgetreten ist, wird dies in der Datei `L_cob_installieren.log` protokolliert.

Wenn möglich, beheben Sie die Fehlerursache und rufen dann zunächst

```
cob_modul_entfernen.x
```

auf, damit evtl. schon installierte Teile entfernt werden (etwaige Fehlermeldungen beim Entfernen können normalerweise ignoriert werden). Dann probieren Sie erneut die Installation mit

```
cob_modul_erzeugen.x.
```

Neben dem Erstellen der Tabellen und Hinzufügen der Prozeduren und Abfragemasken, werden auch Einträge in den Themenbaum und die Tabelle `sachgebiete` gemacht.

Außerdem erhält der User `admin` Zugriffsrechte für das neue Sachgebiet Nr. 27 Kostenrechnung. Die Zugriffsrechte für andere User müssen mit Hilfe der Administrationsabfragen im XML-Frontend oder mit dem Access-Administrationsfronend vorgenommen werden.

2.5 Upgrade eines vorhandenen COB-Moduls

Falls Sie sich nicht sicher sind welche Modulversion Sie nutzen: Die Version steht seit der Version 0.6 in der Textdatei `$SUPERX_DIR/db/module/cob/VERSION`.

Entpacken Sie die neue Version des COB-Moduls. Entladen Sie dann neu aus HISCOB mit dem Entladescript `cob_unload.x`

Starten Sie im Verzeichnis `$SUPERX_DIR/db/module/cob/upgrade`, das Skript `cob_modul1.0rc3_to_rc4.x`. Dieses Skript umfasst auch alle Änderungen der vorhergehenden rc1-rc3-Skripte.

2.6 Laden der COB-Daten nach SuperX

Die entladenen COB-Daten müssen für die Aufnahme nach SuperX im Verzeichnis `$SUPERX_DIR/db/module/cob/rohdaten/unl` stehen. Kopieren Sie die Daten dorthin, oder mounten Sie das Verzeichnis vom COB-Rechner als NFS-Verzeichnis. An der Stelle `$SUPERX_DIR/db/module/cob/rohdaten` müssen außerdem die vom Entladescript automatisch erzeugten Dateien `superx.datum.alt` und `unload.err` stehen.

Um Probleme der Vergleichbarkeit mit HISCOB aufgrund von Gültigkeitszeiträumen zu vermeiden, sollte die Gültigkeit von Kostenstellen,-arten und -trägern überschrieben und die Standbuttons für die Sichten abgeschaltet werden. Erstellen Sie dazu eine Kopie der Datei `finalize.sql.sam` und nennen Sie sie `finalize.sql`.

Zum Übernehmen der Basisdaten nach SuperX wird das Script `$SUPERX_DIR/db/module/cob/cob_update.x --mit-organigramm` gestartet.

(Falls Sie das Organigramm nicht aus der inst-Tabelle von HISCOB übernehmen wollen, lassen Sie den Parameter `--mit-organigramm` weg).

Für den ordnungsgemäßen Betrieb des COB-Moduls ist es wichtig, dass die Kostenstellen aus HISCOB auch im SuperX-Organigramm wiederfinden, daher empfehlen wir die Angabe des Parameters.

Das Skript füllt die Datentabellen; danach werden die Hilfstabellen neu erzeugt.

[Fall Sie die Hilfstabellen einmal manuell aktualisieren wollen, dann starten Sie das Script `$SUPERX_DIR/db/module/cob/hilfstabellen/update_hilfstabellen.x`]

Falls Fehler aufgetreten sind, wird ein Verweis auf eine Logdateien angegeben, die wiederum Verweise auf Logdateien für einzelne Arbeitsschritte enthalten kann. Dadurch muss man vielleicht zwei oder drei Logdateien öffnen, aber nicht in einer einzelnen meterlangen Logdatei umständlich die Fehlerstelle suchen.

Beim regelmäßigen Update wird die Übernahme der COB-Daten über Cronjobs erledigt.

Kopieren Sie dazu `$SUPERX_DIR/db/module/cob_update_cron.x.sam` nach `cob_update_cron.x` und tragen Sie es in die Crontab ein.

Derzeit werden bei jedem Update die gesamten COB-Daten neu übernommen, es ist noch keine Funktionalität zur Übernahme von lediglich geänderten Datensätzen in COB vorgesehen, wie es beim SOS-Modul der Fall ist.

Kommen alle Buchungen nach SuperX?

Die Tabelle `busa` wird nicht komplett in SuperX übernommen; in dieser Tabelle werden alle Buchungen, die für COB nicht relevant sind (Schalter "relevant für COB") herausgefiltert. Der Schalter in der Tabelle `fikr`, `proj` und `inst` wird dazu genutzt. Ebenso werden Storno-Buchungen nicht übernommen. Darüber hinaus werden Buchungssätze, die auf dem Korrekturmonat 13 gebucht sind, auf den Monat 12 gesetzt.

2.6.1 Hochschulspezifische Transformationen

Nach dem Laden der Rohdaten können vor und nach der Transformation hochschulspezifische Anpassungen vorgenommen werden. Dafür sind jeweils die Skripte

```
$COB_PFAD/preparation.sql
```

sowie

```
$COB_PFAD/finalize.sql
```

vorgesehen.

Der SuperX-ETL-Controller (`$SUPERX_DIR/db/bin/module_etl.x`) prüft nach dem Laden der Rohdaten, ob im `$COB_PFAD` ein Skript mit dem Namen `preparation.sql` liegt; wenn ja, dann wird es ausgeführt. Wenn es ein Skript `finalize.sql` gibt, wird dies nach der Datentransformation ausgeführt.

Im COB-Modul liegt bereits eine `finalize.sql` vor. Darin werden die Gültigkeiten von Kostenstellen-, -arten und -trägern zurückgesetzt und das Anzeigen von „Standauswahl-Buttons“ deaktiviert, weil HISCOB regulär noch keine [zeitraumbezogene Auswertung](#) unterstützt.

Wenn Sie die Gültigkeitsangaben in COB gut gepflegt haben und in SuperX zeitbezogene Auswertungen machen wollen, können Sie einzelne Abschnitte auskommentieren, in dem Sie zwei Striche davor setzen.

2.6.2 Separate Übernahme des Organigramms

Wenn das Organigramm aus COB übernommen wird, ist es sinnvoll, den Laderhythmus für das Organigramm anders zu betreiben als den gesamten COB-Update. Während ein COB-Update eher stichtagsbezogen läuft (z.B. 2x pro Semester), sollte das Organigramm eher tagessaktuell vorliegen. Dazu gibt es zum [Entladen](#) (S. 12) ein eigenes Skript (`inst_unload.*`), und zur Übernahme nach SuperX ein Skript namens

```
$COB_PFAD/update_organigramm.x
```

Dieses Script erzeugt aus einer entladenen `cob_inst` ein neues Organigramm. Hochschulspezifische Transformationen können Sie in der Datei `preparation_cob_inst.sql` eintragen (ein Muster mit der Endung `.sam` liegt vor).

Beispiel für hochschulspezifische Transformationen:

```
$COB_PFAD/preparation_cob_inst.sql.sam
```

```
--Manche Hochschulen setzen in ihren Inst-Tabellen die übergeordnete Institution in das Namensfeld, so
```

```
--dass das Organigramm unleserlich wird. Der folgende Update setzt den Namen gleich dem Drucktext.
```

update cob_inst_neu set name=drucktext;

```
--In der cob_inst-Tabelle dient das Feld orgstruktur der Gruppierung von Kostenstellen zu Bereichen.
```

```
--Die Konvention in SuperX ist, dass Lehreinheiten mit 30 gekennzeichnet sind,
```

```
--Fakultäten mit 20, und Institute mit 40. Wenn Ihre Hochschule andere
```

```
--Schlüssel verwendet, müssen diese hier angepasst werden.
```

```
--Wenn Sie z.B. Fakultäten mit 40 verschlüsseln:
```

```
--Zuerst die vorherigen 20er sichern:
```

update cob_inst_neu set orgstruktur=99 where orgstruktur=20;

```
--Dann alle 40er auf 20 setzen.
```

update cob_inst_neu set orgstruktur=20 where orgstruktur=40;

Diese Änderung würde bei Ihnen aktiv, wenn Sie die Datei umbenennen nach `preparation_cob_inst.sql`.

Vor dem Update wird eine Sicherung des Organigramms gemacht. Bei Update-Fehlern wird das alte Organigramm wiederhergestellt, und eine Mail an den Admin geschickt.

Das Script kann allein in die `crontab` eingetragen werden. Wichtig ist, dass Sie im Kopf der Datei `$SUPERX_DIR` richtig setzen.

2.6.3 Setzen von Texten für KLR-Geldgeber

In GOX-GX ist der KLR-Geldgeber ein wichtiges Auswertungsmerkmal. Leider gibt es dafür keine Schlüsseltabelle, so dass Sie dies in SuperX nachpflegen können:

In der Abfrage "Prüfprotokoll Kostenrechnung" befindet sich rechts oben ein Link "KLR-Geldgeber". Wenn Sie (als Administrator) darauf klicken, gelangen Sie in ein Bearbeitungsformular.



Geldgeber (KLR) für das COB-Modul bearbeiten		In diesem Formular können Sie Namen für die dreistelligen <code>klr_geldgeber</code> aus COB vergeben, bzw. nicht mehr benötigte Geldgeber löschen.	
Tid	Apnr	Druck	Lang_1
2.158	D	Drittmittel	KLR-Geldgeber D
2.159	H	Haushaltsmittel	KLR-Geldgeber H

Nach dem ersten Update sind die Schlüssel (z.B. "D" und "H") bereits vorhanden. Sie Texte Druck und Lang_1 können Sie ändern und speichern. Die Änderung ist sofort wirksam.

Sie können auch obsoleete Schlüssel löschen. Ein manuelles Einfügen ist nicht möglich, denn die Schlüssel gelangen automatisch beim COB-Update in die Schlüsseltabelle.

2.6.4 Drittmittel

Jede Hochschule kann über den Schalter "klr_geldgeber" (in COB-GX "Mittelherkunft (KLR)") Geldgebergruppen erzeugen, oben wurde gezeigt wie Sie diese benennen können. Meist existieren landesweite Standards für den zu verwendenden Schlüssel, NRW nutzt z.B. "L", "D" und "P".

Um Buchungen als Drittmittelbuchungen zu kennzeichnen, können Sie die entsprechende Repository-Variablew COB_DRITTMITTEL spezifizieren. In der Abfrage "Prüfprotokoll Kostenrechnung" befindet sich rechts oben ein Link "Filter und Variablen". Wenn Sie (als Administrator) darauf klicken, gelangen Sie in ein Bearbeitungsformular.



Hochschul-Repository	In diesem Formular können Sie Filter im Bereich Kostenrechnung bearbeiten.	Nach Änderungen sollten Sie den Cache aktualisieren SuperX-Manager
-----------------------------	--	---

Variable suchen:

Suche

tid 640		
Variablenname	COB_DRITTMITTEL	Der Variablenname wird in Freemarket-Scripten, beginnend mit "Modulname- "..., z.B. "COB_DRITTMITTEL" genutzt.
Inhalt der Variable	D	Z.B. zum Filter COB_DRITTMITTEL: In einigen Abfragen mit dem Filter "Drittmittel" wird auf die erste Stelle des klr_geldgeber abgefragt, z.B. "D" oder (für Hochschulen in Baden-Württemberg) die "3"
Beschriftung (kurz)	Drittmittel	
Kommentar/Anleitung		
Art der Variable	COB_DRITTMITTEL	
Art der Variable (2)		(Für weitere Gruppierungsebenen)
Sachgebiet	Kostenrechnung	
Sortierschlüssel 1	1	

Im Feld "Inhalt der Variablen" geben Sie die erste Stelle des Schlüssels in `klr_geldgeber` ein, der "Drittmittel" bedeutet. In NRW wäre das z.B. ein "D", in Baden-Württemberg eine "3".

2.7 Vergeben von Rechten für eingeschränkte User

Damit ein Nicht-Administrator mit dem Cob-Modul arbeiten kann, müssen Sie Institutionsrechte und Sichtenrechte vergeben.

Melden Sie sich dazu als `admin` oder `superx` im XML-Frontend an.

2.7.1 a. Institutionsrechte

Wenn Sie einen neuen User anlegen wollen, benutzen Sie die Abfrage *User einrichten*, dort könnten Sie bei *Inst-Rechte* auch angeben, für welche Institution (=Kostenstelle) der User eine Berechtigung erhalten soll.

Wenn der User schon angelegt ist oder Sie weitere Institutionsrechte vergeben wollen, benutzen Sie die Abfrage *User suchen*. Es ist nicht notwendig eine Einschränkung zu machen. Klicken Sie Abschicken. In der Userliste klicken Sie neben dem gewünschten User auf *Bearbeiten*. Auf dem sich öffnenden Formular können Sie neue Institutionsrechte einfügen, in dem Sie unten eine Institution auswählen und *Neue Institution einfügen* anklicken.

Sie können auch vergebene Institutionsrechte entfernen. Markieren Sie die Institution und klicken Sie *Markierte Institution löschen*. Wenn Sie die Gültigkeit der Berechtigung ändern, müssen Sie anschließend *Markierte Institution speichern* anklicken.

2.7.2 b. Sichtrechte

Der User (oder einer der Gruppen zu denen der User gehört) muss Rechte für Sichten auf Kostenstellen, -arten und -träger erhalten.

Sie könnten z.B. eine Gruppe Cob-User anlegen, in dem Sie sich als `admin` oder `superx` im XML-Frontend anmelden und dann die Abfrage *Tabelle suchen* starten. Dann klicken Sie neben der Tabelle *groupinfo* auf *Bearbeiten* und im sich öffnenden Formular auf *neu*. Geben Sie eine *tid* und einen Gruppennamen ein. Zum Abschluß klicken Sie auf *einfügen* und schließen die Fenster.

2.7.2.1 Kostenstellen

Falls noch nicht gemacht, melden Sie sich im XML-Frontend als `admin` oder `superx` im XML-Frontend an.

Rufen Sie die Abfrage *Sicht suchen* mit der Einschränkung *Kostenstellen-Sicht* auf.

Klicken Sie in der Zeile **reguläre Sicht** auf *User- und Gruppenrechte*.

Sicht-Art: **Kostenstellen-Sicht** ; Stand: 01.01.2003

Name	Beschreibung	Art	Bearbeiten	User- und Gruppenrechte
reguläre Sicht	Kostenstellen COB-Import	Kostenstellen-Sicht		

Es öffnet sich ein Formular. Wählen Sie einen User oder eine Gruppe aus und klicken Sie „Neuen User einfügen“ bzw. „Neue Gruppe einfügen“, bis diese in der Liste der User bzw. Gruppen aufgeführt sind, die die Sicht sehen dürfen.

Sichten In diesem Formular können Sie die Le

Name der Sicht	reguläre Sicht
tid	111
Art	Kostenstellen-Sicht
Interner Name	Hauptsicht Kostenstellen
User-Rechte	☒
User, die diese Sicht sehen dürfen:	User Neuen User Neuen User einfügen
Gruppe, die diese Sicht sehen dürfen:	Gruppe ☒ ☐ Markierte Gruppe löschen Neue Gruppe Neue Gruppe einfügen

Nachdem Sie dem User bzw. der Gruppe Rechte für die reguläre Kostenstellensicht gegeben haben, können Sie das Formular schließen und ggfs. weitere Rechte für einzelne alternative Hierarchien vergeben.

2.7.2.2 Kostenträger

Bei Kostenträgersichten sehen eingeschränkte User nur die Kostenträger, die Institutionen zugeordnet sind, für die sie Rechte haben. (intern wird die `cob_proj_to_inst` Tabelle ausgewertet).

Sie können für Kostenträger Sichtrechte nach dem gleichen Vorgehen wie bei Kostenstellen vergeben. Sie können aber auch Rechte für alle Kostenträgersichten auf einmal vergeben, wie im folgenden Abschnitt für Kostenarten beschrieben.

2.7.2.3 Kostenarten

Sie können bei der Vergabe von Rechten für Kostenartensichten genauso vorgehen wie bei Kostenstellen und für jede Sicht einzeln Rechte vergeben, Sie können es sich aber auch einfacher machen und eine Berechtigung für sämtliche Kostenartensichten vergeben.

Rufen Sie dazu die Abfrage *Sicht suchen* mit der Einschränkung „Kosten-/Erlösarten-Sicht“ auf und klicken Sie bei einer Sicht auf *User- und Gruppenrechte* vergeben.

Dann scrollen Sie ans Ende des Formulars. Dort können Sie einzelnen Usern oder Gruppen Rechte für die *ganze Sichtart* geben.

User, die die ganze Sichtart sehen dürfen:	User Neuen User Neuen User einfügen
Gruppen, die die ganze Sichtart sehen dürfen:	User Neue Gruppe Dezernenten Neue Gruppe einfügen

Wählen Sie einen User oder eine Gruppe aus und klicken Sie „Neuen User einfügen“ bzw. „Neue Gruppe einfügen“, bis diese in der Liste der User bzw. Gruppen aufgeführt sind, die die *ganze Sichtart* sehen dürfen.

Das bedeutet, dass der User/die Gruppe *alle* Kostenartensichten sehen darf. Sie brauchen nicht mehr für jede Sicht (alternative Hierarchie) einzeln Rechte vergeben. Auch wenn neue Kostenartensichten dazukommen, erhält der User/die Gruppe bei der nächsten Anmeldung automatisch Rechte dafür.

2.8 Anpassungen der Berichte (nur relevant für NRW)

Das XSL-Stylesheet für das **Berichtsblatt Kennzahlen aus der Kostenrechnung** kann angepasst werden

(`$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/cob/tabelle_html_12000.xml`), es enthält den

Beschreibung SuperX-COB-Modul 1.0

© Projektgruppe SuperX

Briefkopf der Uni Duisburg-Essen. Das zugehörige html-Stylesheet liegt im gleichen Verzeichnis (`tabelle_html_mswf.css`). Hier werden Farben, Schriftarten etc definiert.

2.8.1 Anpassungen des Berichtsblatts Kennzahlen aus der Kostenrechnung (nur NRW)

Das Berichtsblatt betrifft in erster Linie die Hochschulen in NRW. Es besteht aus den Einzelabfragen Kostengrunddaten und Personal nach Landes-/Drittmitteln. Das Stylesheet befindet sich in

`$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_html_11570.xml`.

Sie können diesen Bericht beliebig editieren.

2.9 Entfernen des COB-Moduls

Wenn Sie das COB-Modul nicht mehr benötigen starten Sie das Script

`$SUPERX_DIR/db/module/cob/cob_modul_entfernen.x`.

Dieses Script löscht alle Tabellen, Prozeduren und Abfragen aus der Datenbank, und löscht auch die Einträge im Themenbaum. Danach können Sie den Pfad `$SUPERX_DIR/db/module/cob` löschen.

Wenn Sie nur die Inhalte der Daten- und Hilfstabellen des COB-Moduls löschen wollen (z.B. aus Datenschutzgründen), ohne das ganze Modul zu deinstallieren, können Sie dies mit folgendem Befehl tun:

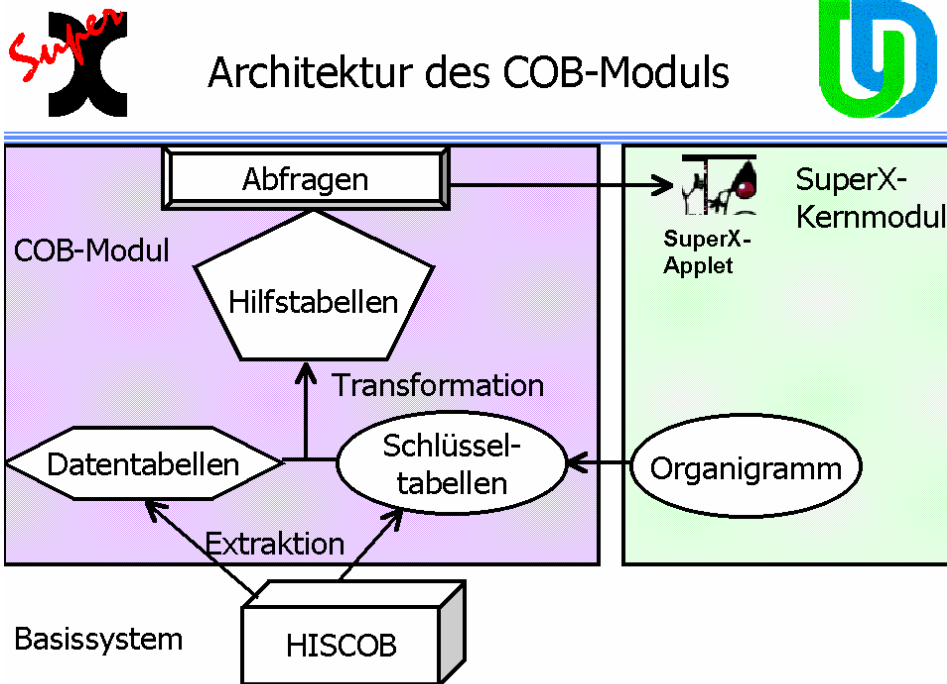
`DOSQL $SUPERX_DIR/db/module/cob/cob_purge_pg.sql` (für Postgres)

bzw.

`DOSQL $SUPERX_DIR/db/module/cob/cob_purge_ids.sql` (für Informix)

3 Bestandteile des COB-Moduls

Im COB-Modul sind die Komponenten von der Datenextraktion bis zur Präsentation enthalten. Die folgende Abbildung zeigt den Datenfluss.



Die Daten werden aus dem Basissystem extrahiert, und die resultierenden Datentabellen werden mit Schlüsseln verknüpft. Daraus werden aggregierte Hilfstabellen erzeugt, die wiederum als Basis für die Abfragen dienen.

3.1 Datentabellen

Die wichtigsten Tabellen des COB- Moduls sind die Grundtabellen

- `cob_busa` <- busa
- `cob_vtbu` <- vtbu
- `cob_umks` <- umks
- `cob_su_imp_stud` <- su_imp_stud

Aus diesen Tabellen werden die wichtigsten Hilfstabellen vom COB-Modul erzeugt.

3.1.1 Tabelle `cob_busa`: Primärbuchungen

Die Tabelle `cob_busa` in SuperX ist ein Auszug der `busa`-Tabelle im COB-System. Das Feld `ch110_institut` ist für eine Verknüpfung zum Organigramm des Kernmoduls vorgesehen. Momentan enthält das Feld den gleichen Wert wie `instnr`.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.1.2 Tabelle cob_vtbu: Verteilbuchungen

Die Tabelle `cob_vtbu` in SuperX entspricht einem Auszug der `vtbu`-Tabelle im COB-System. Darüber hinaus enthält sie wie die Tabelle `cob_busa` die Felder `quell_ch110_inst`, und `ziel_ch110_inst`, die jeweils für Verknüpfungen zum SuperX-Organigramm vorgesehen sind.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.1.3 Tabelle cob_umks: Verzeichnis der Verrechnungssätze/Festpreise

Die Tabelle `cob_umks` in SuperX entspricht einer Kopie der `umks`-Tabelle im COB-System, die das Verzeichnis der Verrechnungssätze/Festpreise darstellt.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.1.4 Tabelle cob_su_imp_stud: Studierendendaten in COB

In der Tabelle `cob_su_imp_stud` werden die Importdaten von HISSOS nach HISCOB gehalten, die dann in der Abfrage **Studierende (gewichtet) für die Kostenrechnung** ausgewertet werden.

Wenn in dieser Abfrage keine Ergebnisse erscheinen, kann dies am automatischen Import von HISSOS nach HISCOB liegen. Bitte beachten Sie, dass Sie vor dem Export von HISCOB nach SuperX in HISCOB die Studierendensummen berechnen: Die Tabelle wird in COB beim Import der SOS Daten nur teilweise gefüllt. Erst beim "Berechnen der Studierendensummen" werden die `stugkey` - Felder gefüllt, die dann in der Auswertung genutzt werden.

Weitere Details zu der Tabelle siehe [Merkmalsbeschreibung](#).

3.2 Schlüsseltabellen

Die Schlüsseltabellen stellen die Metadaten für das COB-Modul dar. Sie sorgen für eine sinnvolle Aggregation der Hilfstabellen. Sie werden direkt von COB übernommen und dürfen nicht manuell nachgearbeitet werden. Nur die Tabelle `cob_inst` dient als Vorlage für das Organigramm und kann falls gewünscht, durch das Skript `preparation_cob_inst.sql` nach Ihren Wünschen angepasst werden. Es ist aber auch möglich, das Kostenstellenverzeichnis aus COB ohne Änderung in das SuperX-Organigramm einzufügen (s.u.).

3.2.1 Die Tabelle cifx

Die Tabelle `cifx` ist Teil des Kernmoduls und wird mit Schlüsseln aus COB gefüllt.

key	hs	Bereich		Bedeutung	Schlüsseltabelle
38	0	COB	SVA	Hochschulnummer	k_hochschule

3.2.2 Die Tabellen cob_cif / cob_cifx

Die Tabellen cob_cif/cob_cifx sind "angereicherte" und nur für COB gültige Varianten der Tabellen cif/cifx mit den zusätzlichen Feldern "ueberkey" und "sort_key"; sie werden mit Schlüsseln aus COB gefüllt. Die Werte für den key entsprechen denen der Kernmodul-cif, COB- interne Schlüssel werden mit key von 700-720 verschlüsselt.

key	hs	Bedeutung	Schlüsseltabelle
30*	<>0	Verzeichnis der Studienfaecher aus HISSOS	k_fach
109*	<>0	Verzeichnis der Verguetungsgruppen	k_bvlgruppe
700*	<>0	Verzeichnis der Flächen-Nutzungsarten nach DIN 277	k_nadin
701*	<>0	Verzeichnis der Kostenflaechenarten	k_kfa
710*	<>0	Schlüsseltabelle fuer (Studien-)Hauptfach/Nebenfach-Kennzeichen	k_kzfa
711*	<>0	Verzeichnis der (SOS-)Studienarten	k_stuart
712*	<>0	Verzeichnis der (SOS-)Studienformen	k_stufrm
713	<>0	Semesterbezeichnung (WS/SS)	zeit
714	<>0	Verzeichnis der Verteilschritte	vari

3.2.3 Die Tabelle cob_inst (Verzeichnis der Kostenstellen)

Die Tabelle cob_inst ist ein Auszug der COB-Tabelle inst, wobei die Felder und Datentypen an das SuperX-Organigramm im Kernmodul angeglichen sind. Hochschulen, die SuperX erstmalig einführen wollen, können also eine vorhandene inst-Tabelle benutzen, um das Organigramm für SuperX zu erzeugen.



Wichtig: Damit der Kostenstellenplan aus der inst-Tabelle in COB übernommen werden kann, muss die Tabelle genau ein root-Element enthalten, das in dem uebinst_nr-Feld ein leeres Element enthält (null oder ""). Dieses Element ist notwendig, um den Baum aufbauen zu können.

Da für den ordnungsgemäßen Betrieb des COB-Moduls die Kostenstellen auch im Organigramm vorkommen müssen, empfehlen wir, beim den Datenupdate des Cob-Moduls mit dem Parameter --mit-organigramm laufen zu lassen. In \$SUPERX_DIR/db/module/cob/cob_update.x --mit-organigramm

Unabhängig vom Update des gesamten Cob-Moduls können Sie das Organigramm aktualisieren mit dem Script

`$$SUPERX_DIR/db/module/cob/update_organigramm.x`

Das Organigramm wird dann aus `cob_inst` gebildet.

Weitere Details siehe [Merkmalsbeschreibung](#).

3.2.4 Die Schlüsseltabelle `cob_fikr` (Verzeichnis der Kosten-/Erlösarten)

Eine Schlüsseltabelle mit dem Kostenartenplan. Diese Tabelle wird aus COB entladen und leicht abgewandelt.

Weitere Details siehe [Merkmalsbeschreibung](#).

Die Schlüsseltabelle enthält zusätzlich das Feld "Hierarchieebene". Sie wird in Abfragen bei der Kostenartenrechnung benutzt.



Wichtig: Damit der Kostenartenplan aus der `fikr`-Tabelle in COB übernommen werden kann, muss die Tabelle genau ein root-Element enthalten, das in dem `ueberg`-Feld ein leeres Element enthält (null oder ""). Dieses Element ist notwendig, um den Baum aufbauen zu können.

3.2.5 Die Tabelle `cob_proj` (Verzeichnis der Kostenträger/Projekte)

Die Tabelle `cob_proj` ist ein Auszug der COB-Tabelle `proj`.



Wichtig: Damit der Kostenträgerplan aus der `proj`-Tabelle in COB übernommen werden kann, muss die Tabelle genau ein root-Element enthalten, das in dem `ueberkey`-Feld ein leeres Element enthält (null oder ""). Dieses Element ist notwendig, um den Baum aufbauen zu können.

Weitere Details siehe [Merkmalsbeschreibung](#).

3.2.6 Die Tabelle `cob_extkotr` (Verzeichnis der externen Kostenträger)

Die Tabelle `k_extkotr` wird regelmäßig mit aus COB entladen und mit dem Script

`$$SUPERX_DIR/db/module/cob/schluesseltabellen/trans_cob_extkotr.sql`

für SuperX angepasst.

Weitere Details siehe [Merkmalsbeschreibung](#).

3.2.7 Die Schlüsseltabelle `cob_kpum` (Verzeichnis der Leistungsarten)

Die Schlüsseltabelle `cob_kpum` enthält das Verzeichnis der Leistungsarten. Die Tabelle stammt aus COB.

Weitere Details siehe Merkmalsbeschreibung ([cob_sx_beschreibung](#)).

3.2.8 Die Schlüsseltabelle `cob_proj_to_inst`

Die Tabelle `cob_proj_to_inst` enthält eine Zuordnung von Projekten zu Institutionen und kommt ebenfalls aus COB. Wenn ein einschränkter User nur bestimmte Institutionen (Kostenstellen) sehen darf, werden bei Kostenträgersichten auch nur die in dieser Tabelle zugeordneten Projekte dargestellt.

Weitere Details siehe [Merkmalsbeschreibung](#).

3.2.9 Die Schlüsseltabelle aggregierung

Die Tabelle `aggregierung` ist eigentlich Bestandteil des Kernmoduls. Für HISCOB werden die relevanten Dimensionen aus der COB-Tabelle `Zeit` übernommen. Die Werte wiederum werden in fast allen COB-Abfragen für den Dialog Zeitraum verwendet.

3.2.10 Die Schlüsseltabelle `cob_drittmittel` (obsolet)

Die Tabelle `cob_drittmittel` wird ohne Änderung aus COB übernommen (`drittmittelherk`) und enthält diejenigen Titel / Kapitel, die Haushaltsbuchungen aus Drittmitteln kennzeichnen.

Beim Update von COB wird diese Schlüsseltabelle neu übernommen. Sie wird für die Kennzeichnung von Drittmitteln verwendet, wenn Quellsystem HISCOB5 ist, oder bei den folgenden HISCOB-Versionen bei Daten von vor 2003 stammen.

Weitere Details siehe Merkmalsbeschreibung.

SuperX geht wie COB davon aus, dass das Kapitel fünfstellig und die Titelgruppe zweistellig ist. Das Haushaltsjahr muss vierstellig angegeben werden.

3.2.11 Die Schlüsseltabelle `cob_stug` (Verzeichnis der Studiengänge)

Die Tabelle `cob_stug` wird ebenfalls ohne Änderung aus COB übernommen und enthält die Studiengänge und deren Regelstudienzeiten, Curricularnormwerte und Lehreinheitsschlüssel. Die Tabelle wird auch für die Erzeugung der Hilftabelle `lehr_stg_ab` genutzt, die Teil des SOS-Modul von SuperX ist.

Weitere Details siehe [Merkmalsbeschreibung](#).

3.2.12 Die Schlüsseltabelle `cob_geldgeber`

Die Tabelle entspricht der Tabelle `gege` in COB.

3.2.13cob_alt_hier, cob_trees, cob_tree_cfg, cob_alt_keys

Die Tabellen cob_alt_hier, cob_trees und cob_tree_cfg werden aus COB für die Darstellung alternativer Hierarchien übernommen.

In cob_alt_keys werden die Tabellen alt_inst, alt_fikr und alt_proj aus HISCOB zusammengefasst.

3.3 Hilfstabellen

Hilfstabellen sind aggregierte Datentabellen, die aus den Basisdatentabellen gebildet werden. Sie erhöhen die Performance der Abfragen, da die Tabellen sinnvoll für einige Abfragen summiert werden.

Wie die COB- Datentabellen werden die Hilfstabellen bei jedem Update komplett neu erzeugt. Je nach Datenvolumen und Rechnerkapazität können sehr unterschiedliche Laufzeiten resultieren. Bei der Installation und für erste Tests sollte deshalb vorsorglich ein abgeschottetes Rechnersystem verwandt werden. Das Script zum Erzeugen der Hilfstabellen lautet

```
$SUPERX_DIR/db/module/cob/hilfstabellen/update_hilfstabellen.x
```

und wird, wenn die Datentabellen ohne Fehler gefüllt werden, im Script

`$SUPERX_DIR/db/module/cob/cob_update.x` gestartet.

3.3.1 cob_busa_aggr

Die Prozedur `sp_cob_busa_aggr` erzeugt aus der Tabelle `cob_busa` eine aggregierte Tabelle mit den Kosten / Erlösen einer Institution bzw. eines Projektes / Studiengangs pro Monat, Jahr und Kostenart. Die Beträge werden je nach Kostenart als Kosten- oder Erlösbuchung summiert.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.3.2 cob_vtbu_aggr

Analog zur obigen Prozedur erzeugt die Prozedur `sp_cob_vtbu_aggr` eine aggregierte Tabelle mit den Verteilbuchungen einer abgebenden und einer empfangenden Institution bzw. Projekts / Studiengangs pro Monat und Jahr.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.3.3 cob_busa_vtbu

Die Prozedur `sp_cob_busa_vtbu` erzeugt aus den Hilfstabellen `cob_busa_aggr` und `cob_vtbu_aggr` eine aggregierte Zusammenstellung der Kosten und Erlöse auf primärer und sekundärer Ebene. Die resultierende Tabelle wird für die Abfrage Kosten und Verteilbuchungen: Schnellinformation verwendet.

Besonderheit NRW:

Wegen den besonderen Anforderungen in NRW wurden die Sekundärkosten auf Ziel-Kostenarten aufgesplittet nach Sekundärkosten auf Ziel-Kostenarten (`vert_ziel`) und Kosten, bei denen das Feld `zfikr` gefüllt ist (`vert_beih`); dies ist sinnvoll für Auswertungen, wo z. B. Beihilfen, die zentral gebucht werden, künstlich den Kostenstellen zugerechnet werden. Die Abfrage Kosten und Verteilbuchungen (inkl. Beihilfen) weist diese Kosten separat aus, in der Abfrage Kosten und Verteilbuchungen (Schnell-Info) werden die Kosten addiert.

Weitere Details siehe Merkmalsbeschreibung (`cob_sx_beschreibung`).

3.4 Alternative Hierarchien / Sichten

In HISCOB angelegt alt. Hierarchien bzw. Auswertungshierarchien (mit eingestelltem Aufklappstatus) werden in den Tabellen `cob_alt_hier`, `cob_alt_keys`, `cob_trees` und `cob_tree_cfg` übernommen.

Um diese im SuperX-COB-Modul verwenden zu können, wird die Sichten-Funktionalität des Kernmoduls benutzt (vergl. Handbuch Kernmodul).

Die Prozedur `sp_cob_alt_hier`, die während des COB-Datenupdates aufgerufen wird, macht Einträge in die Sichtentabelle und zwar für die reguläre „Standardsicht“ auf Kostenstellen-, -arten und -träger als auch für konfigurierte Auswertungshierarchien. Das heißt, dass das nicht alle in COB definierten alt. Hierarchien angezeigt werden, sondern nur solche für die eine „Auswertungshierarchie“ mit spezifischem Aufklappstatus angelegt wurde. Dadurch werden doppelte Anzeigen von alt. Hierarchien vermieden.

Wenn neue alt. Hierarchien aus COB übernommen wurden, müssen sich Benutzer des XML-Frontends neu anmelden, nachdem entweder Tomcat neu gestartet wurde oder der Server-Cache aktualisiert unter <http://rechnername:8080/superx/servlet/SuperXManager>.

3.4.1 Prozedur sp_cob_alt_hier

Die Prozedur sp_cob_alt_hier, die während des COB-Datenupdates aufgerufen wird, macht Einträge in die Sichtentabelle und zwar für die reguläre „Standardsicht“ auf Kostenstellen-, -arten und -träger als auch für konfigurierte Auswertungshierarchien. Das heißt, dass das nicht alle in COB definierten alt. Hierarchien angezeigt werden, sondern nur solche für die eine „Auswertungshierarchie“ mit spezifischem Aufklappstatus angelegt wurde. Dadurch werden doppelte Anzeigen von alt. Hierarchien vermieden.

3.4.2 Prozedur sp_cob_inst_hier

Die Prozedur wird benutzt, um die reguläre Kostenstellen-Sicht für einen bestimmten Stand aufzubauen.

Sie benutzt die Prozedur sp_user_orga aus dem SuperX-Kernmodul, um Rechteeinschränkungen vorzunehmen: welche Kostenstellen darf der User sehen?

Alternative Hierarchien werden Java-intern basierend auf der regulären Sicht mittels der Einträge in der Tabelle cob_alt_keys aufgebaut.

Die tid der ausgewählten Sicht wird standardmäßig als Parameter übergeben, aber bisher noch nicht ausgewertet.

3.4.3 Prozedur sp_cob_ktr_hier

Die Prozedur wird benutzt, um die reguläre Kostenträger-Sicht für einen bestimmten Stand aufzubauen.

Sie benutzt die Prozedur sp_user_orga aus dem SuperX-Kernmodul sowie die Tabelle cob_proj_to_inst um Rechteeinschränkungen vorzunehmen: basierend auf den erlaubten Kostenstellen – welche Kostenträger darf der User sehen?

Alternative Hierarchien werden Java-intern basierend auf der regulären Sicht mittels der Einträge in der Tabelle cob_alt_keys aufgebaut.

Die tid der ausgewählten Sicht wird standardmäßig als Parameter übergeben, aber bisher noch nicht ausgewertet.

3.4.4 Prozedur sp_cob_fikr_hier

Die Prozedur wird benutzt, um die reguläre Kostenarten-Sicht für einen bestimmten Stand aufzubauen.

Rechteeinschränkungen auf einzelne Kostenarten sind bisher nicht vorgesehen.

Alternative Hierarchien werden Java-intern basierend auf der regulären Sicht mittels der Einträge in der Tabelle cob_alt_keys aufgebaut.

Die tid der ausgewählten Sicht wird standardmäßig als Parameter übergeben, aber bisher noch nicht ausgewertet.